NAVAL POSTGRADUATE SCHOOL

Monterey, California

EXTENSION OF THE GRID LINEARIZATION ALGORITHM FOR CONVEX OPTIMIZATION TO NONCONVEX NONLINEAR PROGRAMS

by

James K. Hartman

JULY 1975

Prepared for:
Chief of Naval Research
Arlington, Virginia 22217

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Isham Linder
Superintendent

Jack R. Borsting
Provost

The work reported herein was supported by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

This report was prepared by:

J K Hartman

James K. Hartman
Associate Professor of Operations Research

Reviewed by:

David A. Schrady

DAVID A. SCHRADY, Chairman
Department of Operations Research and Administrative Sciences

Released by:

Robert R Fossum

ROBERT R. FOSSUM
Dean of Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. ~~REPORT NUMBER~~ (14) NPS-55Hh-75071	2. GOVT ACCESSION NO.	~~RECIPIENT'S~~ CATALOG NUMBER (9)
4. ~~TITLE (and Subtitle)~~ (6) Extension of the Grid Linearization Algorithm for Convex Optimization to Nonconvex Nonlinear Programs		5. ~~TYPE OF REPORT & PERIOD COVERED~~ Technical Report, 30 Mar. - 31 Jul 75
		6. ~~PERFORMING ORG. REPORT NUMBER~~
7. AUTHOR(s) (10) James K. Hartman		8. CONTRACT ~~OR GRANT NUMBER(s)~~ (17) ~~61152N~~ RR000-01-10 ~~N0001475WR50001~~
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940 (16) RR000-01		10. PROGRAM ELEMENT, PROJECT, TASK ~~AREA & WORK~~ UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Chief of Naval Research Arlington, VA 22217 (11)		12. ~~REPORT DATE~~ Jul[redacted]74
		13. ~~NUMBER OF PAGES~~ 22
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Arlington, VA (12) 24p.		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Nonlinear Programming, Optimization, Nonconvex programming, Branch and bound, Grid linearization, Separable programing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An algorithm is developed which extends the well known grid linearization method for convex optimization to a class of problems which are not convex. The procedure is a branch and bound method which solves a grid linearization linear program at each stage. Bounds and refinements to the linearization are generated automatically by dual optimizations which involve minimizing single variable nonconvex functions over closed intervals.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601 |

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251 450

I. INTRODUCTION

In this report we will consider the separable nonlinear optimization problem

$$\text{NLP} \qquad \min \quad f(x) \triangleq \sum_{j=1}^{n} f_j(x_j) \qquad (1)$$

$$\text{S.T.} \quad g_i(x) \triangleq \sum_{j=1}^{n} g_{ij}(x_j) \le 0 \quad i=1, \ldots, m$$

$$a_j \le x_j \le b_j \quad j = 1, \ldots, n.$$

One frequently employed algorithm for approximately optimizing (1) is the "Separable Programming" algorithm of Miller [1]. This method forms a piecewise linear approximation to (1) using a fixed grid of points for each x_j and then locally optimizes the approximate program using linear programming.

When all of the functions f_j and g_{ij} are convex functions, then the procedure can be extended to optimize over a variable grid thus giving an arbitrarily accurate approximation in the vicinity of the optimal solution. This procedure, which we will call Grid Linearization is described in Wolfe [2] and can be viewed as an extension of the Dantzig-Wolfe Decomposition Principle for linear programs in that it uses a restricted master linear program to optimize over the current grid at any stage, and subprograms involving the restricted master dual variables to generate new grid points which become new columns of the restricted master at the next stage. For convex programs this procedure has been proved to converge in Dantzig [3]. The procedure is attractive because the restricted master program is a Generalized Upper Bounded L. P. (and hence easy to solve), while the nonlinear subproblems are convex single variable problems (also easy to solve).

In this paper we consider the extension of the grid linearization technique to non-convex separable programs. The result is an algorithm which combines the grid linearization procedure with a branch and bound structure. Restricted master linear programs are solved at each stage of the branch and bound search. Nonconvex single variable subproblems generate bounds for the branch and bound search and also generate new grid points for refinement of the linear approximation.

Section II of this report gives a brief sketch of the grid linearization process, primarily to introduce notation. Section III develops the algorithm for the nonconvex case. Section IV discusses the relation of this method to other algorithms in the literature. Section V deals with computational considerations for attaining efficiency.

II. BRIEF REVIEW OF THE GRID LINEARIZATION METHOD (Convex Case)

A. Fixed Grid

We consider problem NLP (1) with f_j, g_{ij} $(i=1, \ldots, m;$ $j = 1, \ldots, n)$ all convex functions. For each variable $x_j (j=1, \ldots, n)$ suppose we have chosen a (temporarily fixed) grid of points $x_{jk} (k = 0, \ldots, n_j)$ with

$$a_j = x_{jo} < x_{j1} < \ldots < x_{jk} < \ldots < x_{jn_j} = b_j \tag{2}$$

Then any $x_j \in [a_j , b_j]$ can be written as a convex combination of grid points

$$x_j = \sum_{k=o}^{n_j} \lambda_{jk} x_{jk} \tag{3}$$

with

$$\sum_k \lambda_{jk} = 1 \qquad (4)$$

and

$$\lambda_{jk} \geq 0 \qquad (5)$$

If the convex combination is such that for every j at most two λ_{jk} are non-zero and those two have adjacent k indices, then a piecewise linear approximation to $f_j(x_j)$ is defined by

$$\sum_k \lambda_{jk} f_j(x_{jk}) \qquad (6)$$

and similarly for the $g_{ij}(x_j)$.

Thus if we let $f_{jk} = f_j(x_{jk})$ and $g_{ijk} = g_{ij}(x_{jk})$, then the separable programming method defines a linear program P_λ whose decision variables are the convex combination weights λ_{jk} :

$$
\begin{aligned}
P_\lambda \quad & \min \sum_j \sum_k \lambda_{jk} f_{jk} && (7)\\
\text{S. T.} \quad & \sum_j \sum_k \lambda_{jk} g_{ijk} \leq 0 \qquad \forall_i \\
& \sum_k \lambda_{jk} = 1 \qquad \forall_j \\
& \lambda_{jk} \geq 0 \qquad \forall_j , \forall_k .
\end{aligned}
$$

An optimal solution to the linear program P_λ will, if the above adjacency condition is satisfied, provide an optimal solution to the piecewise linear approximation to NLP (1) with solution values given by (3). If NLP is convex, then any feasible P_λ solution gives a feasible NLP solution and it is well known that the optimization process will automatically result in an adjacent solution. Hence P_λ efficiently approximates the solution to NLP.

B. Grid Refinement

When an optimal solution to P_λ has been reached, a natural question is whether there are new grid points in the vicinity of the solution which will improve the piecewise linear approximation and hence the accuracy of the optimal solution. The grid linearization process generates new grid points as follows: Suppose λ is the optimal primal solution and $(\pi, \sigma) = (\pi_1, \ldots, \pi_m, \sigma_1, \ldots, \sigma_n)$ the optimal dual solution to the L. P. P_λ.

A new grid point x_{jk} can improve the P_λ solution only if its reduced cost in the current optimal tableau is negative. This reduced cost is

$$\bar{c}_j = f_j(x_{jk}) - \sum_i \pi_i \, g_{ij}(x_{jk}) - \sigma_j \tag{8}$$

Thus the grid linearization algorithm solves the convex single variable subproblems

$$\min_{a_j \le x_j \le b_j} f_j(x_j) - \sum_i \pi_i \, g_{ij}(x_j) \tag{9}$$

for each $j = 1, \ldots, n$ with solutions $\hat{x}_j$.

If for any j we have

$$f_j (\hat{x}_j) - \sum_{i} \pi_i \, g_{ij} (\hat{x}_j) < \sigma_j \tag{10}$$

then $x_{jk} = \hat{x}_j$ is a new grid point value for x_j which may improve the P_λ solution. If (10) is violated for all $j = 1, \ldots, n$, then the current solution for P_λ translates via (3) into an optimal solution for NLP.

In the Grid Linearization algorithm this process is applied iteratively, alternately optimizing P_λ (called the restricted master problem) and using the subproblems (9) to generate refined grid points and hence to generate new columns which are added to the restricted master P_λ for the next iteration. It has been proved (see for example Dantzig [3]) that this process is infinitely convergent for convex NLP. Further description of the process and proofs of its properties can be found in Lasdon [4] and Dantzig [3] .

III. AN ALGORITHM FOR THE NONCONVEX CASE

A. Lower Bound

For convenience we restate the NLP problem (1)

$$\min f(x) \overset{\Delta}{=} \sum_{j} f_j \, (x_j) \tag{1}$$

$$\text{S.T.} \quad g_i(x) \overset{\Delta}{=} \sum_{j} g_{ij} \, (x_j) \leq 0 \quad i = 1, \ldots, m$$

$$a_j \leq x_j \leq b_j \qquad j = 1, \ldots, n$$

In this section we make no assumptions about convexity of the problem functions. Let

$$C = \{ x \mid a_j \le x_j \le b_j \ , \ \forall_j \} \tag{11}$$

and $C_j = [a_j \ , b_j]$. (12)

Define the Lagrangian function for NLP as

$$\begin{aligned} L(x,\pi) &= \sum_j L_j(x_j \ , \pi) \\ &= \sum_j \left(f_j(x_j) - \sum_{i=1}^{m} \pi_i \, g_{ij} \, (x_j) \right) \end{aligned} \tag{13}$$

and note that it is addititively separable in the variables x_j .

An important lower bound is given in the following Theorem:

<u>Theorem 1</u> Let $\hat{x}$ be any feasible solution for NLP (1) . If

$$\pi_i \le 0, \ i = 1, \ldots, m$$

Then

$$\min_{x_j \in C_j} \ L_j \ (x_j \ , \pi) \le f_j \ (\hat{x}_j) \ . \tag{14}$$

The proof is a standard result from nonlinear duality theory and will not be repeated here.

In particular, Theorem 1 shows that

$$\min_{x \in C} \ L(x, \pi) = \sum_j \min_{x_j \in C_j} \ L_j(x_j \ , \pi) \tag{15}$$

is a lower bound on the (global) optimal objective function value for NLP.

It should be emphasized that 1) this theorem is true for <u>any</u> NLP - no assumptions whatever are required (in particular convexity is not required)

2) The Lagrangian minimization in (15) must be a <u>global</u> minimization. Since $L = \sum L_j$ it suffices to be able to globally minimize the single variable (nonconvex) functions L_j.

3) The bound is tight for well behaved convex programs in the sense that when π is dual optimal, (14) holds with equality $\forall_j$. For non-convex programs, however, it is well known that a "duality gap" may occur so that there may be no feasible x and $\pi \le 0$ for which

$$\min_{x \in C} L(x, \pi) = \sum f_j(x_j)$$

4) The Lagrangian minimization (14) is exactly the same as the Grid Linearization subproblem (9).

B. <u>Feasibility and Optimality in NLP</u>

Suppose we choose a grid of points x_{jk} for each x_j $j=1, \ldots, n$, and set up the P_λ restricted master linear program as in (7). If NLP is a nonconvex program we can no longer guarantee that λ feasible for P_λ implies x given by (3) is feasible for NLP, nor can we guarantee that optimization will automatically lead to an adjacent interpolation.

Nevertheless P_λ is a linearization of NLP. In this section we explore the relation between feasibility and optimality for P_λ and feasibility and optimality for NLP.

Theorem 2 Let λ be primal optimal and (π, σ) be dual optimal for the linear program P_λ (7) with objective function value Z. Let x^* have components $x_j^* = \sum_k \lambda_{jk} x_{jk}$ as in (3). Let $\hat{x}$ (globally) solve

$$\min_{x \in C} L(x, \pi) . \tag{16}$$

If a) $f(x^*) \leq Z$ (17)

b) $g_i(x^*) \leq 0 \quad \forall i = 1, \ldots, m$ (18)

c) $L(\hat{x}, \pi) \geq \sum_{j=1}^{n} \sigma_j$ (19)

Then x^* solves NLP (globally).

Proof at optimality for P_λ we have equal primal L.P and dual L.P. objective function values,

$$\sum_j \sum_k \lambda_{jk} f_j(x_{jk}) = Z = \sum_{j=1}^{n} \sigma_j \tag{20}$$

writing the dual to P_λ shows that dual feasibility requires

$$\pi \leq 0 \tag{21}$$

Thus by Theorem 1, (16) gives

$$L(\hat{x}, \pi) \leq \min \{ f(x) \mid x \text{ feasible for NLP} \} \tag{22}$$

combining (17), (19), (20), (22) with feasibility of x^* in NLP gives

$$Z = \sum_j \sigma_j \leq L(\hat{x}, \pi) \leq \min \{ f(x) \mid x \text{ feasible for NLP} \} \leq f(x^*) \leq Z \tag{23}$$

Thus all the above quantities are equal and x^* solves NLP . QED .

Theorem 2 gives conditions which are sufficient for optimality in NLP. These conditions are not, however, necessary. In particular they will fail to hold for any nonconvex NLP which has a duality gap. The primary value of the theorem is that it suggests an algorithm for getting closer to a solution. When condition c) is not satisfied, then some $\hat{x}_j$ is a new grid point which improves the approximation of P_λ to NLP, while if a) or b) is violated, then the problem is nonconvex in the vicinity of x^* and we must resort to branch and bound. These ideas will be made precise in section III-C.

Theorem 2 required that P_λ possess an optimal solution, but it is also possible that P_λ may be infeasible. The following 2 theorems explore this situation and its implications for the original problem NLP.

<u>Theorem 3</u> If P_λ is infeasible, then the dual to P_λ is unbounded.

<u>Proof</u> The duality theorem of Linear Programming implies that the dual to P_λ is either unbounded or infeasible, but $\pi = 0$, $\sigma_j = \min_k f(x_{jk})$ is a feasible solution to the dual. Hence it is unbounded. QED .

<u>Theorem 4</u> Suppose P_λ is infeasible and $(\pi, \sigma) + \theta(\pi^1, \sigma^1)$ $(\theta \geq 0)$ describes a dual feasible ray along which the dual to P_λ becomes unbounded. Suppose $\hat{x}$ solves the Lagrangian minimization

$$\min_{x \in C} f(x) - \sum_i \pi_i g_i(x) \tag{24}$$

and x^1 solves the related minimization

$$\min_{x \in C} - \sum_i \pi_i^1 g_i(x). \tag{25}$$

If

$$f(\hat{x}) - \sum_i \pi_i g_i(\hat{x}) \geq \sum_j \sigma_j \tag{26}$$

and $$-\sum_i \pi_i^{\,1}\, g_i\,(x^1) \geq \sum_j \sigma_j^{\,1} \tag{27}$$

then NLP is infeasible.

<u>Proof</u> By (24), (25), (26), and (27) we have, for any $x \epsilon C$, and for any $\theta \geq 0$

$$f(x) - \sum_i (\pi_i + \theta\pi_i^{\,1})\, g_i(x) \geq \tag{28}$$

$$\min_{x \epsilon C} \{ f(x) - \sum_i (\pi_i + \theta\pi_i^{\,1})\, g_i(x)\} \geq$$

$$\{\min_{x \epsilon C} f(x) - \sum_i \pi_i\, g_i(x)\} + \theta\{\min_{x \epsilon C} - \sum_i \pi_i^{\,1}\, g_i(x)\} \geq$$

$$\sum_j (\sigma_j + \theta\sigma_j^{\,1}) \, .$$

Thus for any gridpoints $x_{jk} \epsilon C_j$ which we might choose, the resulting P_λ still has $(\pi\ \sigma) + \theta(\pi^1 , \sigma^1)$ as a dual feasible ray along which the dual objective function is unbounded, and hence this P_λ is infeasible. But if $x \epsilon C$ is feasible for NLP, then choosing its components x_j be grid points must give a feasible P_λ . Hence NLP is also infeasible. QED.

C. The Algorithm

The algorithm proposed in this section is a branch and bound method. Branching is done by dividing the interval $C_j = [a_j , b_j]$ for some variable x_j into two subintervals. At each stage t of the search a linearized problem $P_\lambda{}^t$ over some subintervals $C_j{}^t$ is solved. Lagrangian minimizations (14) then provide 1) an optimality test, 2) an infeasibility test, 3) (perhaps) new grid points for incorporation into $P_\lambda{}^t$ as well as 4) a new lower bound on the optimal value of NLP restricted to $x_j \in C_j{}^t$. The detailed description of the algorithm follows:

Step 1 Initialization

For each $j = 1, \ldots, n$ choose an initial grid as $x_{jo} = a_j$, $x_{j1} = b_j$. Let $P_\lambda{}^t$ with $t = 1$(= subproblem counter) be the P_λ program corresponding to this initial grid. Let $C_j{}^t = [a_j , b_j]$. Let $L^t = -\infty$ be the current largest lower bound for $P_\lambda{}^t$. Let $F^0 = +\infty$ be the value of $f(x)$ for the best incumbent feasible solution to NLP found so far. Place $P_\lambda{}^1$ on a list of subproblems and go to step 2.

Step 2 Linear Program

If the list of subproblems is empty, stop. The incumbent solution is global optimal. Otherwise select from the list of subproblems the problem

P_λ^t with the smallest lower bound L^t.

Solve this linear program P_λ^t yielding optimal value Z with optimal primal variables λ and optimal dual variables π, σ.

[If P_λ^t is infeasible the solution yields a dual feasible ray

$$(\pi\ ,\ \sigma) + \theta(\pi^1\ ,\ \sigma^1) \tag{29}$$

along which the dual is unbounded]

Go to step 3.

Step 3 Lagrangian Minimization

Solve the n single variable problems $\min\limits_{x_j \varepsilon C_j^t} L_j\ (x_j\ ,\ \pi)$ giving solutions $\hat{x}_j$.

[If P_λ^t was infeasible also solve $\min\limits_{x_j \varepsilon C_j^t} - \sum\limits_i \pi_i^{\ 1}\ g_{ij}(x_j)$ giving solutions x_j^1.]

Compute $B = \sum\limits_j L_j\ (\hat{x}_j\ ,\ \pi)$.

If $B \geq F^o$ then immediately fathom $P_\lambda{}^t$ and go to step 2.

If $F^o > B > L^t$ then increase the value of the bound for $P_\lambda{}^t$ to $L^t = B$ and go to step 4 . Otherwise go to step 4 without changing the bound.

Step 4 New Grid Points

For each $j = 1, \ldots, n$, if $L_j(\hat{x}_j , \pi) < \sigma_j$ then incorporate $\hat{x}_j$ as a new gridpoint for x_j in the subproblem $P_\lambda{}^t$. Place the new $P_\lambda{}^t$ subproblem on the list and go to step 2. Otherwise go to step 5.

[For an infeasible $P_\lambda{}^t$, if all $L_j(\hat{x}_j , \pi) \geq \sigma_j$ and if

$- \sum \pi_i{}^1 g_{ij}(x^1{}_j) < \sigma_j{}^1$ then incorporate $x_j{}^1$ as a new grid point for $P_\lambda{}^t$

and go to Step 2.

If all $- \sum \pi_i{}^1 g_{ij}(x_j{}^1) \geq \sigma_j{}^1$ also, then fathom $P_\lambda{}^t$ since (by Theorem 4) the corresponding NLP subproblem is infeasible.]

Step 5 Optimality Test

Compute x^* from λ using (3) . If $g_i(x^*) \leq 0$, $i = 1, \ldots, m$, and $f(x^*) < F^o$ then replace F^o with $f(x^*)$ and let x^* be the new incumbent solution.

If a.) $f(x^*) \leq Z$

and b.) $g_i(x^*) \leq 0$ $i = 1, \ldots, m$, then (by Theorem 2) x^* is global optimal for the NLP subproblem over $x \epsilon C^t$. Go to step 2.

If a) or b) is violated go to step 6.

Step 6 Branch

Let $I = \{ i \mid g_i(x^*) > 0 \}$. Let

$$\varphi_j = \sum \lambda_{jk} f_{jk} \quad \text{and} \quad \psi_{ij} = \sum \lambda_{jk} g_{ijk}$$

Let ℓ be the subscript $j = 1, \ldots, n$ which solves

$$\max_{\substack{j=1,\ldots,n \\ i \in I}} \{ f_j(x_j^*) - \varphi_j \; ; \; g_{ij}(x_j^*) - \psi_{ij} \}$$

Let x_ℓ^* be a new grid point for x_ℓ and define two new subproblems,

a.) P_λ^t restricted to $x_\ell \leq x_\ell^*$ (include only grid points to the left of x_ℓ^*)

b.) P_λ^t restricted to $x_\ell \geq x_\ell^*$ (include only grid points to the right of x_ℓ^*)

Let the bound for each of these problems be L^t and place both on the list. Go to step 2 .

IV. RELATION TO OTHER METHODS

The algorithm proposed in Section III is related to several other computational methods for separable nonlinear optimization. One set of relationships can be seen by considering other algorithms whose fundamental mechanism is the grid-linearization representation.

For convex programs P_λ representation with fixed grid is one of the oldest and most used nonlinear programming techniques [1]. The generalization to a variable grid for convex programs [2] is the nonlinear analogue for the Dantzig Wolfe decomposition principle [5].

Nonconvex programs with a fixed grid were considered by Falk [6] and Beale and Tomlin [7] where branch and bound was used to force adjacent interpolations. The current method is the natural culmination of a variable grid and nonconvex problems.

Another set of relationships is with other existing branch and bound methods for nonconvex optimization. A significant contribution here was the work of Falk and Soland [8] and Soland [9] who used convex envelopes of nonconvex functions to form a convex approximating problem which was then imbedded in a branch and bound structure. Our method is similar except that the convex envelope problem is replaced by a sequence of improving P_λ linear approximations. The P_λ problems are easier to formulate and to solve, but they lack the property of being a consistent underestimate of the original problem functions. As a result, the bounds for our problem are derived from Lagrangian duality in contrast to the Falk and Soland bounds which derive directly from the convex envelopes. Greenberg [10] indicates that Lagrangian bounds are stronger than convex envelope bounds in some circumstances. If all the problem functions are concave, then convex envelopes are the same as linear interpolations between the end-points of the intervals C_j^t. In this case our algorithm is very similar to that of Soland [9], and step 4 would never occur. Another similar branch and bound method for the concave case with linear constraints is due to Walkup [11].

V COMPUTATIONAL CONSIDERATIONS

Implementation of the algorithm outlined in section III involves three distinct computational requirements.

a.) Solve the linear programs P_λ^t

b.) Solve the single variable nonconvex Lagrangian minimizations

c.) Generate and maintain the problem list required by the Branch and Bound structure.

In this section we discuss each of these briefly indicating possible choices and tradeoffs which might influence the efficiency of the procedure.

The linear programs P_λ^t which must be solved in step 2 of the algorithm have $m + n$ constraints and as many variables as there are grid points in the subrectangle C^t. The n convexity constraints $\sum_k \lambda_j^k = 1$, $\forall_j$ can be handled implicitly by a Generalized Upper Bounding algorithm, so the effective basis size is only m. Any sparsity in the original NLP constraints $(g_{ij}(x_j) \equiv 0)$ is inherited in the first m constraints of P_λ^t. Thus P_λ^t is a linear program which may have substantial structure and which should not be too difficult to solve. When new grid points are added to an existing P_λ^t the existing solution provides a natural advanced start for the new optimization. Another possibility, when the number of grid points becomes excessive is to drop non-basic grid point columns from the problem. However, for the convex case this destroys the convergence proof.

In Step 3 of the algorithm we must perform the single variable Lagrangian minimization of $L_j(x_j, \pi) = f_j(x_j) - \sum_i \pi_i g_{ij}(x_j)$ over the interval

$x_j \in C_j^t$. If f_j and the g_{ij} are all convex (and $\pi_i \leq 0$) then L_j is convex in x_j and the minimization can be easily handled by methods such as Fibonacci search or perhaps even analytically by setting the derivative to zero. If f_j and the g_{ij} are concave, then L_j is concave also and one endpoint of C_j^t will be minimal. In the general case where L_j is neither convex nor concave, the problem of globally minimizing L_j over an interval is not trivial. Most of the existing methods are heuristic in nature, but if bounds on the derivatives of L_j are known, then a minimax optimal search plan due to Shubert [12] can be used. In any case these are single variable minimizations over an interval and should be substantially easier than a direct n-dimensional search for the solution to NLP.

In most branch and bound algorithms there are tradeoffs between solution strategy and required storage, and these tradeoffs affect the efficiency of the resulting algorithm. This algorithm is no exception. There are two principal tradeoffs to be considered. The first is related to which subproblem on the list should be solved at any given iteration. As the algorithm is written, the "most promising" subproblem (smallest L^t) is attacked at each iteration. It is easy to imagine situations in which two distinct equal valued global minima exist and the algorithm would spend much time switching back and forth between the respective subproblems doing very little work on each at a given iteration. It might be better to do more work on a given subproblem to avoid so many switches even if this means temporarily working on a subproblem which is not the most promising.

The answer to this question depends on how much trouble it is to set up a new subproblem. This depends on how much information is stored about the subproblem and previous solutions to it. Some choices, in decreasing order of storage requirement are the entire tableau, the previous optimal basis inverse, the previous optimal basis vectors, or just the grid points which define the problem. There is clearly a tradeoff here between storage space and solution speed. The decision which is made in any particular case must depend on the computational facilities available and experience with the class of problems to be solved.

VI. AREAS FOR FURTHER STUDY

This report has presented the outline for an algorithm which solves separable nonconvex optimization problems using linear subproblems. The method has close relationships to several existing optimization methods, but also some desirable advantages over them. There are several areas which require further investigation:

a.) At the moment the convergence properties of the method are unknown. An effort to resolve the question is currently underway.

b.) Since the P_λ restricted master problems are linear, it should be possible to exploit special structure in NLP to a considerable degree. We plan to investigate this in the near future.

c.) Computational behavior of the method on particular classes of nonconvex problems is of interest.

d.) As computational experience accumulates the questions of branch and bound organization raised in section V should be resolved.

e.) For fixed grid problems Beale and Tomlin [7] have shown how strong bounds can be derived directly from the P_λ^t optimal tableau. Possible extension to the variable grid case should be investigated.

References

[1] Mitler, C. E., "The Simplex Method for Local Separable Programming", in *Recent Advances in Mathematical Programming*, R. L. Graves and P. Wolfe eds., McGraw-Hill Inc., New York, 1963, pp. 89-100.

[2] Wolfe, P., "Methods of Nonlinear Programming", in *Nonlinear Programming*, J. Abadie, ed., John Wiley and Sons, Inc., New York, 1967, pp. 100-142.

[3] Dantzig, G. B., *Linear Programming and Extensions*, Princeton University Press, Princeton, N. J., 1963, Chapters 22, 24.

[4] Lasdon, L. S., *Optimization Theory for Large Systems*, The Macmillan Co., New York, 1970, Chapter 4.

[5] Dantzig, G. B. and P. Wolfe, "The Decomposition Algorithm for Linear Programming", *Econometrica*, Vol. 9, No. 4, 1961.

[6] Falk, J. E., "An Algorithm for Locating Approximate Global Solutions of Nonconvex Separable Problems", George Washington University, Program in Logistics, Serial T-262, 1972.

[7] Beale, E. M. L., and J. Tomlin, "Special Facilities in a General Mathematical Programming System for Non Convex Problems using Ordered Sets of VAriables", Proceeds 5th International Conference on O. R., 1969.

[8] Falk, J. E. and R. M. Soland, "An Algorithm for Separable Non Convex Programming Problems", *Management Science*, Vol. 15, #9, May 1969.

[9] Soland, R. M., "An Algorithm for Separable Non Convex Programming Problems II - Non Convex Constraints", *Management Science*, Vol. 17, No. 11, July 1971

[10] Greenberg, H. "Bounding Nonconvex Programs by Conjugates", *Operations Research*, Vol. 21, No. 1, January 1973.

[11] Walkup, D. W., "On a Branch and Bound Method for Separable Concave Programs", Boeing Scientific Research Laboratories, Mathematical Note No. 527, September 1967.

[12] Shubert, B. O., "A Sequential Method of Seeking the Global Maximum of a Function", *SIAM Journal of Numerical Analysis*, Vol. 9, No. 3, September 1972.

INITIAL DISTRIBUTION LIST

	No. of Copies
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
Dean of Research Code 023 Naval Postgraduate School Monterey, California 93940	1
Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
Dr. Thomas Varley Office of Naval Research Arlington, Virginia 22217	2
Library, Code 55 Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	5
Professor Donald P. Gaver Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
Professor James K. Hartman Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	40